Implementing a formal model of inflectional morphology

Benoît Sagot, Géraldine Walther

Implementing a formal model of inflectional morphology

Benoît Sagot[1] Géraldine Walther[2,3]

(1) Alpage, INRIA & Univ. Paris-Diderot, 75013 Paris
(2) Laboratoire de Linguistique Formelle, CNRS & Univ. Paris-Diderot, 75013 Paris
(3) UFR de Langue Française, Univ. Panthéon-Sorbonne, 75005 Paris
benoit.sagot@inria.fr, geraldine.walther@linguist.jussieu.fr

Abstract. Inflectional morphology as a research topic lies on the crossroads of many a linguistic subfield, such as linguistic description, linguistic typology, formal linguistics and computational linguistics. However, the subject itself is tackled with diverse objectives and approaches each time. In this paper, we describe the implementation of a formal model of inflectional morphology capturing typological generalisations that aims at combining efforts made in each subfield giving access to every one of them to valuable methods and/or data that would have been out of range otherwise. We show that both language description and studies in formal morphology and linguistic typology on the one hand, as well as NLP tool and resource development on the other benefit from the availability of such a model and an implementation thereof.

1 Introduction, motivation and related work

Contrarily to syntax and derivational morphology, inflectional morphology has the advantage of dealing, for a given language, with a finite range of data. Given a set of lexical units, it is possible to list all their inflected forms. From a theoretical point of view, one can therefore expect any formal approach to inflectional morphology to account not only for the data, i.e., inflectional paradigms in a given language, but also for the regularities and irregularities found within them.

Because of its finiteness, inflectional morphology also readily lends itself to typological approaches, where regularities and irregularities can be studied in a contrastive way. Among those approaches, the corpus of work carried out in the framework of **canonical typology** [1] aims at modeling and explaining inflectional phenomena accross languages, including **non-canonical** phenomena such as syncretism, suppletion, heteroclisis or defectivity.

The confined set of data underlying inflectional morphology makes for the perfect place to combine approaches as different as computational linguistics, formal linguistics, linguistic typology, and descriptive linguistics, i.e., approaches that seldom get to combine in a global enterprise of precise language description, analysis and effective processing. The work described in this paper aims at furthering the combination of those complementary approaches. We describe the development of a lexical framework redesigned for implementing a theoretical

and formal approach to inflectional morphology as well as improving the quality and speed of lexical resource and tool development. As a framework designed for all of the subfields cited above, it entails specific benefits for each one of them, but its main advantage lies in the combination of its possible different outcomes.

More specifically, this paper describes Alexina$_{\text{PARSIL}}$, a formalism for encoding inflectional descriptions (lexicon and grammar) that aims at filling the gap between morphologically and typologically motivated approaches on the one hand and implemented approaches on the other hand, as will be discussed in the remainder of this section. Indeed, Alexina$_{\text{PARSIL}}$ is both:

- an **implementation formalism for** PARSIL, a formal model of inflectional morphology [2, 3] that accounts for concepts underlying the canonical approach of morphological typology.[1] We briefly describe the last version of PARSIL, on which Alexina$_{\text{PARSIL}}$ relies, in Section 2. In particular, we point out the major innovations with respect to earlier versions of PARSIL [2, 4];
- an **extension of the Alexina lexical framework** [5] used in the field of Natural Language Processing (NLP) for modeling lexical information and developing lexical resources. The morphological layer of the (original) Alexina formalism is sketched in Section 3.

In Section 4, we show how we extended the morphological components of Alexina for turning it into an implementation formalism for PARSIL, namely Alexina$_{\text{PARSIL}}$. Finally, in Section 5, we show why the Alexina$_{\text{PARSIL}}$ formalism and tools have been greatly beneficial to works both in descriptive and formal morphology, in particular in studies about Latin passivisation and Maltese verbal inflection and in studies comparing the compacity of morphological descriptions, as well as in NLP, for the efficient development of a large-scale and linguistically sound morphological lexicon for German.

1.1 A tool for enhancing studies in theoretical linguistics

From the point of view of theoretical linguistics in general and linguistic typology in particular, the main goal in the study of inflectional morphology lies in the description and comparison of inflectional systems belonging to different languages. As mentioned above, joint efforts therefore entail the following advantages. In order to simply generate paradigms from a morphological description of a given language, to describe and measure regularities and irregularities in these paradigms, or even to perform cross-linguistic comparisons, only formal and computational approaches can lead to reliable results: Formalisation allows for guaranteeing the consistency of an analysis, in particular within a full morphological system; Implementation allows for concretely verifying the validity

[1] We use here the term 'implementation' for expressing the fact that Alexina$_{\text{PARSIL}}$ provides a way to create and manipulate electronic resources (lexicon, grammar) that follow morphological analyses developed within the PARSIL formal model of inflectional morphology. Alexina$_{\text{PARSIL}}$ is both a language and a set of tools that can process a morphological description written in this language, e.g., for generating an automatic inflection tool.

of the analysis; In addition, large-scale implementation allows for a verification of the quasi-exhaustivity of the proposed analysis. In particular, it is a way to assess the overall relevance of a complete morphological description and the relative importance of a given phenomena within the full morphological system.

Yet, this formalisation and implementation approach is still rarely used in theoretical morphology. In many cases, formalisations are somewhat approximative, and sometimes only concern the modeling of one particular phenomenon, often independently of the overall morphological system it has been extracted from [3]. Few models exist for which real implementations are available, that make it possible to validate theoretical assumptions. Among these few models are PFM [6] and *Network Morphology* [7], together with Finkel's *Cat's Claw* for the former[2] and the `DATR` formalism for the latter [8] and its extensions such as `KATR` [9]. Still, large-scale implementations in these frameworks remain rare. For example, analyses available on the *Cat's Claw* web site rarely involve more than fifty lexical entries. One exception is Brown and Hippisley's analysis of Russian nouns [7], which involves 1,500 lexical entries.[3]

On the other hand, computational approaches, most of them based on finite-state automata [10], have no difficulty for efficiently generating correct paradigms. As a matter of fact, it has been shown by Karttunen [11] that if one reduces morphological theories, including PFM and *Network Morphology*, solely to their ability to generate paradigms, they come down to realisational systems equivalent to finite-state automata [10]. However, even if computational approaches perfectly achieve this goal, they are often criticised, in the eyes of theoricists, for lacking what is the most interesting aspect from the theoretical point of view, namely explictely modeling regularities and irregularities within paradigms. We introduce a means to easily implement formal analyses in a typologically sound framework that benefits from the data processing power available through computational approaches alone. [4] On an experiment carried out on modelling Maltese verbal inflection, we show the benefit for formal approaches to rely on computational approaches.

1.2 Improving the quality and efficiency in NLP resource development

On the other side of the scope, the issue for computational linguistics, and in particular NLP tool and resource development, lies in rapidly building high

[2] *Cat's Claw*: `http://www.cs.uky.edu/~raphael/linguistics/claw.html`.

[3] This analysis is avaiable at `http://networkmorphology.as.uky.edu`.

[4] As a result, our contribution does not rely in the computational aspects *per se*, and are complementary to standard finite-state morphology tools. Drawing a parallel with syntax, one could compare Alexina with the LKB plateform [12], corresponding to the HPSG theoretical model of syntax [13]. On the other hand, finite-state tools such as XFST [10] or FOMA [14] would correspond to optimized and generic parser generators for context-free grammars such as the Lex/Yacc pair (http://dinosaur.compilertools.net).

quality resources. We show on experiments on German, that Alexina$_{\text{PARSLI}}$ also allows for quickliy setting up new NLP ressources that are theoretically sound.

2 The PARSLI model of inflectional morphology

As indicated by its acronymic name, PARSLI [2, 3], 'PARadigm Shape and Lexicon Interface', is a formal model of inflectional morphology in general and of the interface between the shape of a lexeme's paradigm and the structure of its lexical entry in particular. This **inferential-realisational** model in the sense of Stump [6] has been built as a formalisation of the notions developed within the typological framework of canonical typology [1]. It explicitely models regularities and irregularities within a paradigm and/or an inflectional system, so called **non-canonical phenomena** as defined within canonical typology. Among those are suppletion [15], heteroclisis [16], deponency (or morphosyntactic mismatches) [17], defectiveness [18], overabundance [19], *etc.* It relies on the explicit representation of each non-canonical phenomenon as pieces of information directly encoded within the structure of a specific lexical entry. Moreover, the extent of the non-canonical phenomenon can be quantified with specific non-canonicity measures developed within the framework [3].

A preliminary version of the framework had been introduced in [2] and used in experiments described in [4]. The version described here is the one presented in [3]. Compared to the version in [2], it contains many innovations, among which the formal representation of the lexical entry itself, including the specification of an entry's morphosyntactic feature structure set, the formalisation of an **inflectional category**, the layered representation of inflection and the encoding of the full range of non-canonical phenomena defined within canonical typology.

2.1 Representation of a lexical entry

PARSLI explicitly formalises the notion of inflectional lexical entry. As shown in Figure 2, each entry (or lemma) is defined through a phonological base input I-PHON for the realisation rule sequences, its inflectional category I-CAT (such as *verb*, *transitive verb*, *noun etc.*), a set of expressable morphosyntactic feature sets, a set of suppletive stems S-STEM or forms S-FORM and an inflectional pattern I-PAT consisting of a set of subpatterns.

2.2 Morphosyntactic feature sets and inflectional categories

Each lexical entry is defined through its membership within a specific inflectional category. Each one of those categories *canonically* expresses a certain set of morphosyntactic feature sets. Latin nouns, for example, will express two number values (SG and PL) as well as five differents case values (NOM, ACC, GEN, DAT and ABL). If a lexeme belongs to a specific category, it will canonically express the same set of features and be marked as *standard* in its lexical entry (see the feature set under MSF for French BALAYER 'swipe' at Figure 2). Sometimes,

lexemes will express more or less features than expected. These deviations will be noted under MSF in their lexical entry. They are then considered to display the non-canonical phenomena of **overabundance** [19], resp. **deficiency** [3].

2.3 Realisation zones

One of ᚹᚪᚱᛊᛚᛁ's major innovations with respect to comparable models [6, 20, 7] is the generalisation of the notion of paradigm partition in the sense of Pirelli and Battista's 'partition spaces' [21] or Bonami and Boyé's thematic spaces [22] to the exponence [23] level by stipulating so called **realisation zones**, illustrated by the different colours in Figure 1. Instead of associating lexical entries with a complete inflection class, ᚹᚪᚱᛊᛚᛁ associates every entry with realisation zones that contain the realisation rules allowing for the construction of specific partitions of this lexeme's paradigm. These realisation rules can thus be combined in different ways to account for the realisation of different types of paradigms. In particular, heteroclite paradigms as illustrated by the Slovak data in Table 1, can be accounted for by the combination of zones usually used by lexemes belonging to two different inflection classes. The Slovak data shows how some animal nouns use the singular inflection zone of animate nouns to build their singular forms, and the plural zone of inanimate nouns to build their plural forms. ᚹᚪᚱᛊᛚᛁ defines realisation classes, such as inflection class Z^{exp}_{anim} for masculine animate nouns in Table 1, as default combinations of realisation zones: classes are thus a derived notion built from the clustering of realisation zones that are observed in the construction of a significant number of paradigms. The significance itself is derived from a notion of decriptive economy: classes are only stipulated if this allows for a more compact representation of the whole system.

The set of morphosyntactic feature sets expressed by the realisation rules of a given realisation zone is called this zone's **partition space**.

Table 1. Heteroclite Slovak animal nouns

	Z^{exp}_{anim}: MASC. ANIMATE CHLAP 'guy'		Z^{exp}_{inan}: MASC. INANIMATE DUB 'oak tree'		MASC. HETEROCLITE OROL 'eagle'	
	$z^{exp}_{\text{anim,sg}}$: SG	$z^{exp}_{\text{anim,pl}}$: PL	$z^{exp}_{\text{inan,sg}}$: SG	$z^{exp}_{\text{inan,pl}}$: PL	$z^{exp}_{\text{anim,sg}}$: SG	$z^{exp}_{\text{inan,pl}}$: PL
NOM	*chlap*	*chlap-i*	*dub*	*dub-**y***	*orol*	*orl-**y***
GEN	*chlap-a*	*chlap-ov*	*dub-a*	*dub-ov*	*orl-a*	*orl-ov*
DAT	*chlap-**ovi***	*chlap-om*	*dub-u*	*dub-om*	*orl-**ovi***	*orl-om*
ACC	*chlap-a*	*chlap-ov*	*dub*	*dub-**y***	*orl-a*	*orl-**y***
LOC	*chlap-**ovi***	*chlap-och*	*dub-e*	*dub-och*	*orl-**ovi***	*orl-och*
INS	*chlap-om*	*chlap-mi*	*dub-om*	*dub-mi*	*orl-om*	*orl-ami*

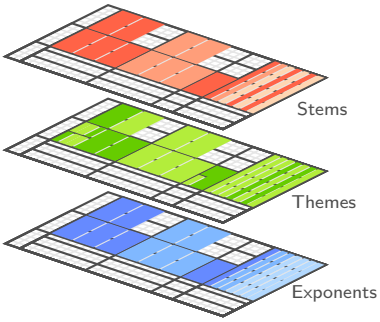

Fig. 1. Inflectional layers

BALAYER	
I-PHON	balayer
I-CAT	*verb*
MSF	{ *standard* }
S-STEM	*(empty)*
S-FORM	*(empty)*
I-PAT	$(z^{s}_{ay}, \text{id}), (z^{exp}_{v1,1}, \text{id})$
	$(z^{s}_{ay}, \text{id}), (z^{exp}_{v1,2}, \text{id})$
	$(z^{s}_{ai}, \text{id}), (z^{exp}_{v1,2}, \text{id})$

Fig. 2. French BALAYER 'swipe'

2.4 A layered representation of inflection

PARSIL relies on a highly structured representation of form realisation by specifying various inflectional layers as illustrated in Figure 1. Every realisation zone belongs to a specific layer. Among these layers is at least one stem layer dealing with potential stem alternations (allomorphy). But other, optional theme (green) or exponence (blue) layers can also be stipulated. Figure 1 illustrates a realisational architecture containing three layers – one of each type. The theme and exponence layers could also further be split into several layers if needed.

The realisation of a given form thus consistst in the application of one realisation rule per layer. For example, inflecting the French verb BALAYER 'swipe' consists in applying realisation rules of two different layers, one stem layer and one exponence layer. One specificity of French verbs in *-ayer* like BALAYER is that they are overabundant [19] for half of their paradigm, i.e., half of their paradigm's cells display two different forms for expressing the same morphosyntactic feature set: *balayent* and *balaient* are both valid for expressing the features 3.PL.PRS.IND, whereas only *balayons* holds for 1.PL.PRS.IND. More precisely, each of the overabundant cells can be filled using one of two different stems (*balay-* and *balai-*) but the same exponents. As a result, we use two different stem zones, corresponding to two different realisation rules, and split the set of realisation rules into two exponence zones: zone $z^{exp}_{v1,1}$ corresponding to overabundant cells and $z^{exp}_{v1,2}$ for non-overabundant cells.[5] For example, building the form *balayent* for 3.PL.PRS.IND involves the application of the stem realisation rule in stem zone z^{s}_{ay}, which builds the stem *balay-*, followed by a form realisation rule from zone $z^{exp}_{v1,1}$ that adds the suffix *-ent*. Generating the alternate form *balaient* involves the other stem realization rule from stem zone z^{s}_{ai}, which builds the stem *balai-*, followed by the same inflection zone $z^{exp}_{v1,1}$ at the exponence layer. On the other hand, *balayons* obtains its 1.PL.PRS.IND suffix *-ons* from a rule in $z^{exp}_{v1,2}$, applied

[5] In the name $z^{exp}_{v1,1}$, 'v1' stands for 'First group verb', whereas the final '1' is the index of the exponence zone.

on the stem *balay-* generated by the stem realisation rule in z^s_{ai}. The combination of z^s_{ai} with $z^{exp}_{v1,2}$ is not allowed, therefore **balaions* is not generated.

The licit associations of realisation zones across layers are stated in a lexeme's inflectional subpatterns grouped together in the inflectional pattern I-PAT, as indicated in the lexical entry for BALAYER illustrated by Figure 2.

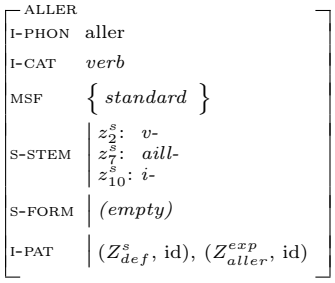

Fig. 3. French ALLER 'to go'

Fig. 4. Serbo-croatian BRAT 'brother' (data from [24])

2.5 Suppletive stems and forms

Lexical entries can also specify suppletive stems or forms. In the canonical case, the list of suppletive stems S-STEM or forms S-FORM will be marked as empty (see the example of BALAYER). But in the case of a verb like French ALLER 'to go', suppletive stems can be specified along with a stem index corresponding to the stem zone's partition space in which the suppletive stem is used. The three suppletive stems *v-*, *aill-* et *ir-* of ALLER are listed under S-STEM in Figure 3. Suppletive forms are listed under S-FORM along with the feature set they express.

2.6 Realisational couples and transfer rules

For each lexical entry, the inflectional pattern I-PAT specifies a certain number of subpatterns consisting of realisational couples, such as the couple (Z^s_{reg}, id) in the entry of Serbo-croatian BRAT 'brother'. These couples themselves consist in a realisation zone or class such as Z^s_{reg} and a **transfer function**. Transfer functions are identity functions in the canonical case. However, there are cases where a feature expressed by a lexeme's form differs from the feature canonically expressed by a given realisation rule. Such a case arises for example in nouns like BRAT who build their plural form by using realisation rules usually used for building singular forms. Such nouns specify a particular transfer function such as t_{NB} that allows for specifying the morphosyntactic mismatch between the features expressed (PL) and the features realised (SG).

This summary presentation of the PARSEME model will be extended while showing how we implemented the formal notions within Alexina$_{\text{PARSEME}}$ (section 4).

3 The original Alexina formalism

We have based our implementation of the ParSeL model on the Alexina lexical formalism [25–27, 5]. Alexina covers both the morphological and the syntactic level, only the former being relevant here.[6] Alexina's original morphological layer, although significantly different, shares some fundamental properties with ParSeL. These include in particular an explicit interface between the inflectional lexicon and the inflectional grammar.

Alexina already has a good track record as a lexical formalism, as there exist a number of medium- and large-scale lexicons for diverse languages (see Table 2), among which the first, largest and richest is the French lexicon Le*fff* [26, 5]. Indeed, the development of Alexina lexicons is facilitated by the availability of associated development, maintenance, validation and extension tools and interfaces. Moreover, all Alexina lexicons are freely available (including Alexina$_{\text{ParSeL}}$ lexicons). This guarantees that morphological analyses and lexical data can be checked and used by anyone, be it for typological, morphological or NLP studies.

Table 2. Alexina lexicons. Darker lines correspond to Alexina$_{\text{ParSeL}}$ lexicons. References for each of these resources can ben found in [5, 3]

LEXICON	LANGUAGE	#LEMMAS	#LEXEMES	#FORMS	#DISTINCT FORMS
Le*fff*[7]	French	120,000	125,000	550,000	460,000
Le*ff*e	Spanish	180,000	180,000	1,500,000	700,000
Le*ff*ga	Galician	70,000	70,000	750,000	500,000
Le*ff*la	Latin	2,200	2,200	115,000	96,000
EnLex	English	350,000	350,000	580,000	510,000
DeLex	German	63,000	63,000	2,100,000	405,000
PolLex	Polish	240,000	240,000	1,400,000	360,000
SkLex	Slovak	50,000	50,000	470,000	250,000
PerLex	Persian	30,000	30,000	550,000	460,000
KurLex	Kurmanji Kurdish	22,000	22,000	410,000	240,000
SoraLex	Sorani Kurdish	520	520	30,000	25,000
MaltLex	Maltese	560	560	9,000	7,200

The way Alexina encodes morphology explicitely relies on a paradigmatic approach. Each lexical entry is associated with an inflection class, as illustrated in the upper part of Figure 5 with five verbal lexical entries from the Le*fff*.[8] Each **intensional entry** consists of a citation form (respectively *accoutumer* 'accustom', *appeler* 'call', *enrichir* 'enrich', *dormir* 'sleep', *admettre* 'admit') and

[6] Alexina also entails a means to represent derivational morphology, but this also lies beyond the scope of this paper.

[7] The Alexina$_{\text{ParSeL}}$ version of the Le*fff*'s morphological description is called NEW in [4]. Its original ("official") version relies on the original Alexina formalism [5].

[8] For the sake of simplicity, syntactic information is not displayed.

an **inflection class**. In the morphological grammar, each inflection class is explicitely defined through the realisation rules that describe how it will build paradigms. In Figure 5, the inflection classes involved are `v-er` for the regular and productive class of first group verbs, `v-ir2` for the regular and virtually non-productive class of second group verbs, `v-ir3` for third-group verbs in *-ir* and `v55` for one of the irregular third-group inflection classes. For some lexical entries, the inflection class is associated with inflection class **variants**, which allow for selecting specific rules for generating some of the forms in the paradigm. For example, in Figure 5, variants `dbl` and `std` respectively apply to first-group verbs which double their stem-final consonant in some cells (cf. *appeler* / *appelle*) and to first-group verbs which do not (cf. *peler* 'peel' / *pèle*). The lower part of Figure 5 shows a few inflected entries, or **extensional entries**, generated by the lexical entries of the upper part. The lexemes' morphological categories are displayed next to the inflected form, along with their citation forms and a morphological tag encoding the feature sets expressed by the inflected form.

```
accoutumer    v-er:std
appeler       v-er:dbl
enrichir      v-ir2
dormir        v-ir3
admettre      v55
```

```
accoutuma      v     accoutumer      J3s
accoutume      v     accoutumer      PS13s
accoutumant    v     accoutumer       G
appela         v     appeler         J3s
appelle        v     appeler         PS13s
appelant       v     appeler         G
enrichit       v     enrichir        J3s
enrichit       v     enrichir        P3s
enrichissant   v     enrichir        G
dormit         v     dormir          J3s
dort           v     dormir          P3s
dormant        v     dormir          G
admit          v     admettre        J3
admet          v     admettre        P3
admettant      v     admettre        G
```

Fig. 5. Lexical entries from the Le*fff*: intensional entries in the upper part, a few corresponding extensional entries in the lower part

An Alexina morphological grammar contains two main sections: (1) a set of morphonological rules — or rather, as all Alexina lexicons to date are orthographic, morphographemic rules that simulate morphonological rules; and (2) the morphological part proper, i.e., the description of each inflection class.

The morphonological part usually starts by the definition of graphemes (including digraphs or trigraphs) and grapheme classes (e.g., the set of back vowels). These classes can then be used when defining morphonological rules.

In the original Alexina formalism, the strictly morphological part of the grammar defines inflection classes by specifying realisation rules for inflected forms associated with the corresponding feature tag. These rules can only involve suffixation and/or prefixation. Any other morphological operation must therefore be simulated in two steps, namely first an affixation rule that inserts the necessary information for the subsequent application of dedicated "morphonological" rules that produce the correct output.[9] A realisation rule can also simply stipulate the form for a given morphological tag to be realised in the same way that the form for another tag (this corresponds to Stump's [6] "rules of referral"). This constitutes a simple (and directional) modeling of the notion of syncretism. An inflection class can also inherit all or some rules from another inflection class. Thus, in the Le*fff*, the inflection class `adj-4` for adjectives that inflect both in gender (*-e* for the feminine) and in number (*-s* in the plural) inherits all rules from the inflection class `nc-4` for masculine nouns inflecting in both gender and number using the same suffixes (e.g., *doctorant/doctorante* 'PhD student'). All types of realisation rules (explicit or inheritance) can be restricted based on their input, using positive (`rads=`) or negative (`rads_except=`) regular-expression-like **constraints**.[10]

At the technical level, an Alexina morphological grammar is an XML document. A dedicated tool can then compile into an inflection script (which can inflect the associated intensional lexicon), a "disinflection" (ambiguous lexicon-free lemmatisation) tool and a derivation tool (that produces all possible derived lexemes based on regular derivation patterns, sketched above but not described here). This technical architecture is preserved in Alexina$_{\text{PARSL}}$.

4 Adapting Alexina to PARSL: Alexina$_{\text{PARSL}}$

Our development of an implementation formalism for PARSL is based on the orignal Alexina formalism, which we adapted for it to take into account PARSL-specific concepts. The notions of inflection class variant, classes of letters, morphonological rules and constraints on realisational rules have been retained from Alexina, and sometimes generalised.

As already mentioned, a partial Alexina implementation of a preliminary version of PARSL was used in [4]. However, the work presented in this section goes far beyond, for at least two reasons.

[9] Cf., producing *appelle* from *appeler* and *jette* from *jeter* in a unified way entails using a non-strictly concatenative operation: the duplication of the stem-final consonant. There, one can use an affix such as *-2e* followed by a "morphonological" rule that rewrites *t_ 2* as *tt_* and *l_ 2* as *ll_* ("_" indicates a morph boundary).

[10] E.g., it is possible to posit two rules for a same tag, and specify that one applies only to stems ending with a consonant or a glide, the other only to stems ending with a vowel or a glide. As a result, the corresponding cell will be overabundant for lexical entries whose stem ends in a glide, yet for those lexical entries only.

First, the PARSLI formalism itself has been heavily enriched since [4]. The first version of PARSLI was for example restricted to one stem level and one exponence level, and was not able to deal with the full range of non-canonical phenomena. Because this was sufficient for encoding French verbal inflection, it allowed for carrying the compacity experiments described in [4]. The latest version of PARSLI, whose implementation within Alexina is described here, has proven adapted to a large variety of typologically diverse languages since, such as Sorani Kurdish (Indo-European, western Iranian), German (Indo-European, Germanic, see Section 5.2) Latin (Indo-European, Italic), Maltese (western Semitic), Khaling (Sino-Tibetan, Kiranti), and others.

Second, the implementation of the preliminary version of PARSLI used in [4] only covered those PARSLI notions that were required for implementing the four morphological descriptions of French verbal inflection used in the compacity study. Notions such as transfer rules were already in PARSLI but not implemented. Moreover, Alexina$_{\text{PARSLI}}$ offers many ways to simplify morphological grammars, that are also new to the work presented here. This includes for example (1) various factorisation mechanisms; (2) a mechanism for specifying, for each category the inventory of morphological attributes, their values, and their incompatibilities, thus specifying the inventory of cells in the paradigms; (3) a novel and generic way to model morphological (realisational) operations; (4) the possibility to encode realisation rules using *rule blocks*, in a way similar to Stump's [6] Paradigm Function Morphology.

4.1 Morphological operations

As mentioned above, the only available operations for expressing realisation rules in the original Alexina formalism were prefixation and suffixation operations, and more complex operations had to be simulated *via* morphonological rules. Alexina$_{\text{PARSLI}}$ now allows for modeling non-concatenative operations. More specifically, it is now possible to define in the morphological grammars all morphological operations required by realisation rules, including non-concatenative ones. Indeed, such morphological operations are considered as specifically belonging to the morphological system of the language at hand. Suffixation (written as `append=`) and prefixation (`left_append=`) are retained from Alexina as basic morphological operations. In addition, `insert` operations allow for inserting segments, while `replace` allows for replacing one segment by another. These basic operations can be used for defining more complex operations, as illustrated in Figure 7, which we explain below. The definition of morphological operations, just as morphonological rules in the original Alexina, often makes use of the notion of *letter class*, sketched above and retained within Alexina$_{\text{PARSLI}}$. Two letter class definitions from MaltLex are snown in Figure 6.

Based on letter classes and on the primitive operations (`append`, `left_append`, `insert` and `replace`), operations such as shown in Figure 7 can be defined. In this Figure, the operation `deleteV1` defined for Maltese stems stipulates for example that for a given stem construction rule input (`source=`) with a CVCVC structure, applying `deleteV1` produces a new stem (`target=`) with a CCVC

```
<letterclass name="C" letters="b␣ċ␣d␣f␣ġ␣g␣għ␣ħ␣h␣j␣k␣l␣m␣n␣p␣q␣r␣s␣t␣v
    ␣w␣x␣ż␣z␣’␣"/>
<letterclass name="V" letters="a␣e␣i␣o␣u␣ie"/>
```

Fig. 6. Two letter classes from MaltLex

structure. A `replace` operation keeps the occurrences of a letter class `c` unchanged if it is written `[:c:]` on both sides of the rule, as is the case for all consonants `[:C:]` in the definition of `deleteV1`, and deletes it if it is written `[0:c:]`, as is the case for the first vowel `[0:V:]`. Within an operation definition, the first applicable rule is applied, and the following rules are ignored (except if the applied rules specifies explicitely the opposite, with `stop="0"`). If none of the rules can be applied on a particular input, the operation fails.

Erasing segments is not the only option when using `replace`. Not preserving a segment can also be used as the first part of a real replacement operation. Thus, any occurrence of the symbol "`_`" in the output of a rule will be replaced by an argument given to the operation in addition to its input, as is the case in the second operation definition in Figure 7. For example, one can invoke the `deleteV1changeV2` operation in a realisation rule by writing `deleteV1changeV2(i)`, in order to replace the second vowel by an *i*. Note that each occurrence of "`_`" in the output of a rule corresponds to a different argument to be given to the operation at invocation time.

```
<!-- Maltese (MaltLex) -->
<operation_definition name="deleteV1">
  <replace source="[:C:][0:V:][:C:][:V:][:C:]"
           target="[:C:][:C:][:V:][:C:]" />
  <replace source="[:C:][0:V:][:C:][:V:]" target="[:C:][:C:][:V:]" />
</operation_definition>
<operation_definition name="deleteV1changeV2">
  <replace source="[:C:][0:V:][:C:][0:V:][:C:]"
           target="[:C:][:C:]_[:C:]" />
  <replace source="[:C:][0:V:][:C:][0:V:]" target="[:C:][:C:]_" />
</operation_definition>

<!-- Latin (Leffla) -->
<operation_definition name="redup-initial">
  <replace source="#[1:C:][0:V:]" target="#[1:C:]_[1:C:]_"/>
</operation_definition>
```

Fig. 7. Morphological operations in Alexina (data from MaltLex and Leffla)

Finally, it is also possible to duplicate segments, as necessary for encoding reduplication. The last operation definition in Figure 7 is an example of a reduplication operation, used for producing several Latin verbal stems. The initial consonant is indexed by a numeric identifier, here 1 in `[1:C:]`, which allows for invoking it more than once anywhere in the rule. In this operations, it is reduplicated in the output. In addition, the initial vowel is dropped. This operation is also an illustration of a two-argument operation: the first, respectively second occurrence of "`_`" in the output of the rule will be replaced by the first, respectively second argument provided to `redup-initial` upon invokation. For example, calling `redup-initial(e,e)` on the stem *fall-* from the verb FALLO 'deceive' outputs the stem *fefell-*.

4.2 Stem allomorphy, stem suppletion and form suppletion

In the original Alexina formalism, each lexical entry is meant to have one unique stem. Accounting for **stem allomorphy** requires to circumvent this limitation by using available mechanisms in ways that are both very *ad hoc* and not linguistically sound.[11] In Alexina$_{\text{FRMORLI}}$, stem allomorphy is directly handled by two distinct mechanisms, one for regular stem allomorphy (as in Iranian languages, for example) and one for irregular stem allomorphy (as for the verb *aller* 'go' in French, illustrated in Figure 3.[12]

Regular allomorphy is modeled using realisation rules at the stem level within the grammar. Figure 8 provides two such rules for modeling Maltese regular stem allomorphy, which is illustrated the two paradigms in Table 3. Within Maltese paradigms, stem allomorphy involves up to six distinct stems (RAD1 to RAD6). The examples in Table 3 involve four stems, indicated by four distinct cell background colors (two of the stems are syncretic with others: RAD1 and RAD2 on the one hand, and RAD5 and RAD6 on the other hand).

The other way to account for multiple stems in Alexina$_{\text{FRMORLI}}$ is to provide them explicitely in the lexical entry. It is the natural way to deal with irregular stem allomorphy, i.e., cases where stem creation is not dealt with by the grammar but constitutes a lexical irregularity of the lexical entry at hand. If one considers that French verbs have twelve stems (often syncretic), following Bonami and Boyé [22] and retaining the same stem identifiers from stem 1 to stem 12, each of these stems corresponds to a slot after the `/` symbol in the lexical entry, sorted and separated by a comma. For example, in the case of ALLER, as shown in Figure 9, suppletive stems specified in the lexicon are stem 2 *va-*, stem 7 *aill-* and stem 10 *i-*. Other stems are deduced from these stems or from the citation form's stem *all-* by regular stem allomorphy rules. For instance, stem 3 is deduced by syncretism with stem 2, and so on.

[11] For example, one can define "morphonological" rules that play the role of stem generators. In that case, the realisational rule can suffix the stem with a special marker that will later be interpreted by these "morphonological" rules. This temporary solution was used, for example, for simulating stem allomorphy Persian.

[12] We refer the reader to [3] for a discussion on why and how regular and irregular stem allomorphy should be distinguished from one another.

Table 3. Paradigms for the Maltese verbs RASS and MESS

		RASS 'press'	MESS 'touch'
RAD2	PFV 1.SG	*rasséjt*	*messéjt*
	PFV 2.SG	*rasséjt*	*messéjt*
	PFV 1.PL	*rasséj.na*	*messéj.na*
	PFV 2.PL	*rasséj.tu*	*messéj.tu*
RAD1	PFV 3.M.S	*rás*	*méss*
RAD3	PFV 3.F.S	*rás.set*	*més.set*
RAD4	PFV 3.PL	*ras.sé:w*	*mes.sé:w*

perfective sub-paradigms

		RASS 'press'	MESS 'touch'
RAD5	IPFV 1.SG	*nróss*	*nmíss*
	IPFV 2.SG	*tróss*	*tmíss*
	IPFV 3.M.S	*jróss*	*jmíss*
	IPFV 3.F.S	*tróss*	*tmíss*
RAD6	IPFV 1.PL	*nrós.su*	*nmís.su*
	IPFV 2.PL	*trós.su*	*tmís.su*
	IPFV 3.PL	*jrós.su*	*jmís.su*

imperfective sub-paradigms

```
<table name="CVCC" rads="[:C:][:V:][:C:][:C:]">
  <item name="S1"/>
  <item name="S2" source="S1" append="ej"/>
  <item name="S3" source="S1" operation="" />
  <item name="S4" source="S1" append="e"/>
  <item name="S5" source="S1" operation="changeV1(o)" rads="[:C:]a[:C
      :][:C:]"/>
  <item name="S5" source="S1" operation="changeV1(i)" rads="[:C:]e[:C
      :][:C:]"/>
  <item name="S5" source="S1" operation="changeV1(i)" rads="[:C:]i[:C
      :][:C:]"/>
  <item name="S6" source="S5" operation="" />
</table>
```

Fig. 8. Regular stem allomorphy (MaltLex data, after Camilleri and Walther [28])

```
aller      v:23r/,va,,,,,aill,,,i
dire       v:3re/dis,,di,,,,,,,,,dit/2.pl.prs.ind=dites
```

Fig. 9. Irregular stem allomorphy and form allomorphy (data from a modified Le*fff*)

The original Alexina formalism had no way to encode **form suppletion**. Again, this non-canonical phenomena had to be modeled in a non-satisfying way.[13] In Alexina$_{\text{PARSLI}}$, it is possible to list suppletive forms in a lexical entry. They override any form the morphological grammar might want to generate. Figure 9 illustrates this on the example of the verb *dire* 'say', which has the irregular 2.PL.PRS.IND *dites* instead of the regular *disez*.

In the case of overabundant suppletive forms, the mechanism already available in Alexina is preserved: they can simply be listed explicitely as such. For

[13] Either by assigning to the entry an inflection class that would not generate forms for all cells, if any, and specifying the missing forms explicitely; or by considering (almost) the whole form as an exponent (suffix) over an (almost) empty stem.

```
<!-- Maltese (MaltLex) -->
<level type="stem" level="1">
    <partitionspace name="S1" features="3.m.sg.pfv"/>
    <partitionspace name="S2" features="1.pfv|2.pfv"/>
    <partitionspace name="S3" features="3.f.sg.pfv"/>
    <partitionspace name="S4" features="3.pl.pfv"/>
    <partitionspace name="S5" features="sg.ipfv"/>
    <partitionspace name="S6" features="pl.ipfv"/>

<!-- Latin (Leffla) -->
<level type="exponent" level="3">
   <partitionspace name="I1" features="ipfv.ind|ipfv.sbjv|prs.inf"/>
   <partitionspace name="I2" features="pfv.ind|pfv.sbjv|pst.inf"/>
   <partitionspace name="I3"
             features="prs.ptcp|fut.ptcp|fut.inf|sup|pst.ptcp|grv|grd"/>
```

Fig. 10. Definition of partition spaces (data from MaltLex and Le*ff*la)

example, MARRON 'brown' is listed in the Le*fff* as inflecting for number (*marrons*). In addition, the Le*fff* lists an additional plural form (*marron*).

4.3 Inflectional layers, zones and patterns

Another new feature of Alexina$_{\text{PARSLI}}$ is that it implements inflectional layers. The original Alexina formalism could deal with one exponence layer only. The implementation of the preliminary version of PARSLI used in [4] was dealt with one stem layer and one exponence layer only. In Alexina$_{\text{PARSLI}}$, a description can involve an unbounded amount of layers (`level`): Zero or one stem layer (`type="stem"`), zero to many theme layers (`type="theme"`) and zero to many exponence layers (`type="exponent"`).

PARSLI defines partition spaces as subsets of feature structure sets. They constitute one of the ways to refer to realisation zones as defined in PARSLI. In Alexina$_{\text{PARSLI}}$, these partition spaces can be defined on a per-layer basis. We illustrate this mechanism in Figure 10 on Maltese for the stem layer and on Latin for the exponence layer.

One of the main innovation in PARSLI, and therefore in Alexina$_{\text{PARSLI}}$, is the introduction of **realisation zones**. In Alexina$_{\text{PARSLI}}$, they can be defined in two different ways. Either directly using XML tags within a given level, or by invoking them as the intersection of a realisation class and a partition space. For space reasons we do not illustrate this here, but the morphological grammars in MaltLex and Le*ff*la, which are freely available, contain many examples thereof.

In Alexina, intensional entries are associated with inflection classes. Alexina$_{\text{PARSLI}}$ implements PARSLI's view, according to which a lexical entry is associated with realisation zones through a **pattern**. Inflection classes are only a secondary notion: they emerge as observable generalisations that capture sets of zones often

used together by lexical entries. Inflection classes are indicated by `table` tags in Alexina, zones by `zone`. As a result, lexical entries in the intensional lexicon are associated with patterns, which are defined in the grammar. A pattern contains at least one *subpattern*, which is defined in turn as a set of realisation zones (`realzone`),[14] one per realisational layer (see Figure 11).[15] Each subpattern can only produce either zero or one form for a given feature set (i.e., for a given cell). Regular overabundance therefore requires patterns that contain several subpatterns. In addition, each pattern is provided with a *morphological category*. This allows for computing the inventory of cells that has to be filled by the pattern. How and from which information this inventory is computed is explained in the next section. Figure 11 illustrates how patterns are defined in the grammar based on Latin verbal data from Le*ff*la.

```
<pattern name="v-aA" cat="v" >
  <subpattern>
    <realzone level="1" table="s-reg"/>
    <realzone level="2" table="a"/>
    <realzone level="3" table="v-A"/>
  </subpattern>
</pattern>
<pattern name="v-aAB" cat="v" >
  <subpattern>
    <realzone level="1" table="s-reg"/>
    <realzone level="2" table="a"/>
    <realzone level="3" table="v-B"/>
    <realzone level="3" partitionspace="I3" table="v-A"/>
  </subpattern>
</pattern>
```

Fig. 11. Examples of pattern definitions (data from Le*ff*la)

4.4 Morphosyntactic features and definition of paradigms' cells

In the original Alexina formalism, morphosyntactic features appeared only as tags associated with (exponence) realisation rules. In , a realisational

[14] As mentioned above, a realisation zone is either invoked as such, or by proving a realisation table and a partition space. In the latter case, if the partition space is omitted, the whole table is considered as a zone. In addition, a transfer function, in the sense sketched above, can be specified.

[15] In fact, constraints can be associated with a `realzone`, such as a partition space for which the rule is valid, one or more *variants* that must be assigned for the lexical entry in order for the rule to apply, or constraints on the input of the rule (`rads=` and `rads_except=`). Therefore, several `realzone`s can be used in the same subpattern for the same level. This is one of the factorisation devices mentioned above.

model, each form is considered as the realisation of a morphological feature structure. Alexina$_{\text{FARSI}}$ therefore explicitely models feature structures. The inventory of cells specific to a given morphological category is computed based on a unification mechanism. For a given morphological category (`category`), the cells to be realised are obtained as the combination of all attribute-value pairs that are mutually compatible. Therefore, an Alexina$_{\text{FARSI}}$ morphological grammar specifies for each category the inventory of attributes, each possible value for each of these attribute, as well as exclusion rules (e.g., this value for this attribute is incompatible with that value for that attribute, or with the attribute itself; or this particular feature structure is invalid).

Finally, morphological feature structure sets can be defined, and then associated with lexical entries for encoding deficiency. For example, in French, impersonal verbs are associated with the set `impers`, which only contains cells from the verbal paradigm whose morphological feature structure unify with 3.SG.

4.5 Realisation rules

Above, we have described how morphological operations are defined, and can be invoked (including by realisation rules). Contrarily to the orignal Alexina, Alexina$_{\text{FARSI}}$ associates realisation rules with morphological feature structures. At a given layer, a realisation rule will be applied if its feature set successfully unifies with the feature set of the form being generated.

In Alexina$_{\text{FARSI}}$, it is possible, as in PFM, to defined rule blocks within a zone or a table. In each block, one and only one rule applies given an input feature set. In the Maltese example in Figure 12, that illustrates the unique exponence table in MaltLex, we make use of two rule blocks `block="1"` and `block="2"`. The first one realises aspect and person, the second one realises number (suffix *-u* for plural forms). As in PFM, Alexina$_{\text{FARSI}}$ allows for writing *portemanteau* rules that span over more than one adjacent blocks, and have precedence over standard rules. In our example, `block="1-2"` allows for one rule to short-cut both blocks. Last, if more than one rule can apply for generating the same form, the first one is used (contrarily to PFM, which would use the most specific one).

5 Use cases

5.1 Alexina$_{\text{FARSI}}$ for language description

Throughout the previous section, we have used examples from MaltLex, a lexicon that covers the semitic-based part of the Maltese verbal system. The first version of this description was based on Camilleri's analysis [29], which had been formalised and implemented in Alexina$_{\text{FARSI}}$. The lexicon associated with this implementation was a quasi-exhaustive inventory of 600 semitic-based verbal entries, extracted from the list of 850 first-binyan verbs from Spagnol's *Maltese Language Resource Server* [30] by manually filtering out incorrect entries. This implementation, including the lexicon, has shown that Camilleri's analysis correctly accounts for most data on first-binyan stems. However, it has also unveiled

```
<level type="exponent" level="3">
  <table name="exponence" rads="">
    <item block="1-2" suffix="na" features="1.pl.pfv"/>
    <item block="1-2" suffix="et" features="3.f.sg.pfv"/>
    <item block="1" suffix="t" features="1.sg.pfv|2.pfv"/>
    <item block="1" prefix="n" features="1.ipfv"/>
    <item block="1" prefix="t" features="2.ipfv"/>
    <item block="1" prefix="t" features="3.f.sg.ipfv"/>
    <item block="1" prefix="j" features="3.ipfv"/>
    <item block="2" suffix="u" features="pl"/>
  </table>
</level>
```

Fig. 12. Realisation rules (MaltLex data)

that several phenomena were not handled, including instances of overabundance. Moreover, as it is only a model of stem alternation, it does not account for the behavior of the extension vowel, which appears in the imperfective forms of some verbs. The extension of the model for taking into account this extension vowel was greatly eased by its implementation, which allowed for generating all inflected forms and validate them. In other words, the ᑭᗩᖇᔕᒪ formalisation and the Alexina$_{\text{ᑭᗩᖇᔕᒪ}}$ implementation, together with a large-scale lexicon (for the class of verbs at hand) were of a crucial help for correcting and extending an analysis previously assumed as complete, thus contributing to improve the understanding of Maltese verbal morphology [28].

5.2 Alexina$_{\text{ᑭᗩᖇᔕᒪ}}$ for developing lexical resources

The development of lexical resources for NLP can also benefit from a morphologically sound model of inflection. An example thereof is the recent development of DeLex, a new Alexina morphological lexicon for German. Apart from DeLex, and to our best knowledge, there is surprisingly no freely available morphological lexicon for German, as pointed out by Adolphs [31].

German morphology is not stricly concatenative, in particular because nominal and verbal inflection involves vowel alternations (*ablaut* and *umlaut*) at the stem level, leading to stem allomorphy. In addition, overabundance is massive, in particular within nominal and adjectival paradigms, both at the stem and at the exponence levels.[16] As a result, the manual development of a morphological grammar for German was simplified and speeded up thanks to notions defined in ᑭᗩᖇᔕᒪ and implemented in Alexina$_{\text{ᑭᗩᖇᔕᒪ}}$. For example, our morphological grammar involves two realisational levels for adjectives and nouns, namely one

[16] E.g., at the exponence level, many masculine and neuter nouns can bear the suffix *-s* or *-es* for the GEN.SG, and/or the null suffix or the suffix *-e* for the DAT.SG. The stem level also shows overabundance for these nouns, specifically for the plural stem.

stem level and one exponence level. This allows for defining only one adjectival exponence table, as all variation within adjectival inflection lies at the stem level, i.e., in the way the comparative and superlative stems are built[17] or resort to morphonology. Verbs involve an additional exponence level, which uses the unique adjectival exponence zone for inflecting the past participle.

The efficient development of this morphological grammar has been realised together with the extraction of lexical data and continuous validation of both the grammar and the lexicon *via* the paradigms they generate. Indeed, we extracted large-scale lexical information from the German Wiktionary[18], which provided us with non-structured, noisy and incomplete raw data. The morphological grammar proved very useful for detecting and correcting this raw data.

The current version of DeLex now contains as many as 63,000 intensional entries (citation form + inflection pattern) generating over 2 million extensional entries (inflected form + citation form + morphological feature structure), which cover 405,000 distinct inflected forms.

5.3 Alexina$_{\text{PARSIL}}$ for quantitative formal morphology

Beyond language description and analysis, a large-scale implementation (large-coverage grammar and medium- or large-scale lexicon) is a prerequisite for carrying out quantitative linguistic studies. A preliminary version of PARSIL had already proven sufficient to implement four different descriptions of French verbal inflection [4]. The associated implementation allowed for objectively comparing these descriptions on the basis of a dedicated information-theoretic measure [4].

New complexity assessment tools have been developed, which are now compatible with Alexina$_{\text{PARSIL}}$ as presented in this paper. They have been used for quantitatively assessing various descriptions of Latin and Maltese verbal inflections, including the descriptions of L*eff*la and MaltLex [3] used as examples throughout this paper. These comparisons have shed new light on formal morphological issues such as the balance between heteroclisis and deponency in Latin, or the boundary between morphonology and (autonomous) morphology in Maltese.

6 Conclusion

In this paper we have introduced the latest version of Alexina$_{\text{PARSIL}}$, an implementation of the PARSIL model for inflectional morphology that extends the Alexina lexical framework. We have shown on examples from various languages the relevance of the model and its implementation. Our aim is now, thanks to Alexina$_{\text{PARSIL}}$, to strengthen our efforts towards joint work between various specialists of morphology, be they descriptive linguists, typologists, formal linguists or computational linguists.

[17] For example, stem suppletion (*gut* 'good', *besser* 'better' *best-* 'best'), stem-related deficiency (*alkoholfrei* 'alcool-free' has no comparative or superlative) or stem overabundance (*frei* 'free', *freier* 'freer', *freist-* or *freiest-* 'freest').

[18] `http://de.wiktionary.org`

References

1. Corbett, G.G.: Agreement: the range of the phenomenon and the principles of the Surrey database of agreement. Trans. of the Philological Society **101** (2003) 155–202
2. Walther, G.: Measuring morphological canonicity. In Perko, G., ed.: Linguistica. Volume 51. Faculté des Arts, Université de Ljubljana, Ljubljana, Slovénie (2011) 157–180 Internal and External Boundaries of Morphology.
3. Walther, G.: Sur la canonicité en morphologie — Perspective empirique, formelle et computationnelle. PhD thesis, Université Paris-Diderot (2013)
4. Sagot, B., Walther, G.: Non-canonical inflection : data, formalisation and complexity measures. In Mahlow, C., Piotrowski, M., eds.: Systems and Frameworks in Computational Morphology. Volume 100., Zurich, Suisse, Springer (2011) 23–45
5. Sagot, B.: The L*efff*, a freely available, accurate and large-coverage lexicon for French. In: Proceedings of LREC'10, Valletta, Malta (2010)
6. Stump, G.T.: Inflectional Morphology. A Theory of Paradigm Structure. Cambridge University Press, United Kingdom (2001)
7. Brown, D., Hippisley, A.: Network Morphology: A Defaults-based Theory of Word Structure. Cambridge University Press (2012)
8. Evans, R.P., Gazdar, G.: Inference in DATR. In: Proceedings of EACL'89. (1989) 66–71
9. Finkel, R., Stump, G.T.: Generating Hebrew verb morphology by default inheritance hierarchies. In: Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages, Philadelphia, PA, United States (July 2002)
10. Beesley, K.R., Karttunen, L.: Finite State Morphology. Studies in Computational Linguistics. CSLI Publications (2003)
11. Karttunen, L.: Computing with realizational morphology. In Gelbukh, A., ed.: Computational Linguistics and Intelligent Text Processing. Volume 2588 of Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Germany (2003) 205–216
12. Copestake, A.: Implementing Typed Feature Structure Grammars. CSLI Publications, Stanford, CA, United States (2002)
13. Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. CSLI Publications, Stanford, CA, United States (1994)
14. Hulden, M.: Foma: a finite-state compiler and library. In: Proceedings of EACL (demos). (2009) 29–32
15. Boyé, G.: Suppletion. In Brown, K., ed.: Encyclopedia of Language and Linguistics (2nd ed.). Volume 12. Elsevier, Oxford, United Kingdom (2006) 297–299
16. Stump, G.T.: Paradigm function morphology. In Brown, K., ed.: Encyclopedia of Language and Linguistics. Elsevier, Oxford, United Kingdom 171–173
17. Baerman, M.: Morphological reversals. Journal of Linguistics **43** (2007) 33–61
18. Baerman, M., Corbett, G.G., Brown, D., eds.: Defective Paradigms: Missing Forms and What They Tell Us. Oxford University Press (2010)
19. Thornton, A.: Overabundance (multiple forms realizing the same cell): A non-canonical phenomenon in italian verb morphology. In Martin Maiden, John Charles Smith, M.G., Hinzelin, M.O., eds.: Morphological Autonomy: Perspectives From Romance Inflectional Morphology. Oxford University Press (2011)
20. Stump, G.T.: Heteroclisis and paradigm linkage. Language **82** (2006) 279–322
21. Pirrelli, V., Battista, M.: The Paradigmatic Dimension of Stem Allomorphy in Italian Verb Inflection. Italian Journal of Linguistics (2000) 307–380

22. Bonami, O., Boyé, G.: Supplétion et classes flexionnelles dans la conjugaison du français. Langages **152** (2003) 102–126
23. Matthews, P.H.: Morphology. Cambridge University Press, United Kingdom (1974)
24. Baerman, M.: Deponency in Serbo-Croatian Typological Database on Deponency, Surrey Morphology Group, CMC, University of Surrey.
25. Sagot, B.: Automatic acquisition of a Slovak lexicon from a raw corpus. In: Lecture Notes in Artificial Intelligence 3658 (© Springer-Verlag), Proceedings of TSD'05, Karlovy Vary, Czech Republic (2005) 156–163
26. Sagot, B., Clément, L., de La Clergerie, E., Boullier, P.: The Le*fff* 2 syntactic lexicon for French: architecture, acquisition, use. In: Proceedings of LREC'06, Lisbonne, Portugal (2006)
27. Sagot, B.: Building a morphosyntactic lexicon and a pre-syntactic processing chain for Polish. In: Proceedings of LTC'05, Poznań, Poland (2007) 423–427
28. Camilleri, M., Walther, G.: What small vowels and a large lexicon tell us about Maltese verbal inflection (2012) Presentation at the 8th Décembrettes. Bordeaux, France.
29. Camilleri, M.: Island morphology: Morphology's interactions in the study of stem patterns. Linguistica **51** (2011) 65–84
30. Spagnol, M.: A Tale of Two Morphologies. Verb structure and argument alternations in Maltese. PhD thesis, University of Konstanz, Constance, Germany (2011)
31. Adolphs, P.: Acquiring a poor man's inflectional lexicon for german. In: Proceedings of LREC'08, Marrakech, Maroc (2008)